Docket No. AUS920030742US1

LOAD BALANCING TO SUPPORT TAPE AND DISK SUBSYSTEMS ON SHARED FIBRE CHANNEL ADAPTERS

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention relates to storage area networks and, in particular, to multi-path input/output in a storage area network. Still more particularly, the present invention provides a method, apparatus, and program for load balancing to support tape and disk subsystems with shared paths in a multi-path input/output environment.

2. Description of Related Art:

A storage area network (SAN) is a network of storage devices. In large enterprises, a SAN connects multiple machines to a centralized pool of disk storage. Compared to managing hundreds of servers, each with its own storage devices, a SAN improves system administration.

In multiple path input/output (MPIO), there is a plurality of routes or connections from one specific machine to one specific device. For example, with a logical disk device on a redundant array of independent disks (RAID), the accessing host uses a Fibre channel (FC) adapter connected to an FC switch, and the FC switch in turn is attached to the RAID array. There may be eight, or as many as thirty-two or more, FC adapters in both the host and the device and there may be many more FC switches in the SAN fabric.

Considering a SAN with eight adapters in the host and the device, if each host adapter is connected to a device adapter through a switch, then there may be eight paths from the host to the device. If the switches are interconnected, then there may be many more paths from the host to the device. Path management software chooses paths to be used for each device.

Attaching a tape subsystem and a disk subsystem to the same Fibre channel adapter is currently not supported, because a tape subsystem achieves optimum performance with a dedicated path from the host. In order to perform write operations on a tape, the tape must spin. If data stops, the tape must stop spinning and a rewind operation must be performed to reposition the tape to wait for more data. As such, a tape device operates best with a consistent flow of data to be written.

A tape subsystem generates an underflow error when the amount of data in a buffer drops below a predefined threshold. The tape subsystem also generates an overflow error when the amount of data in the buffer exceeds another threshold. The problem of I/O starvation occurs due to the sequential streaming of tape storage technology. Tape subsystems with large caches are as susceptible to I/O starvation as subsystems with smaller caches.

If the path to the tape subsystem is shared and resources are being used by competing storage subsystems, the amount of input/output (I/O) to the tape subsystem may decrease and result in underflow errors due to I/O

starvation. This may lead to a backup operation failing and having to be restarted, which may result in timeout errors. Therefore, current systems dedicate one adapter to the tape subsystem.

However, since the adapter dedicated to the tape subsystem is not utilized most of the time, the host is not able to efficiently utilize all of the paths from all of the adapters. When an adapter is dedicated to the tape subsystem, a large number of paths are also dedicated to that subsystem.

SUMMARY OF THE INVENTION

The present invention recognizes the disadvantages of the prior art and provides a mechanism load balancing to support tape and disk subsystems with shared paths in a multi-path input/output environment. The present invention provides a mechanism for monitoring I/O activity of each device and the total I/O activity for each adapter. When there is low I/O activity for the tape subsystem, the I/O for the disk subsystems may be spread across all available adapters and paths. When I/O activity for the tape subsystem increases, the I/O activity for the disk subsystems may be reduced on the adapter processing the tape I/O, but will continue across all other adapters. If the tape subsystem begins to report errors due to I/O starvation, the disk I/O activity may be adjusted until the errors stop.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 depicts a pictorial representation of a storage area network in which the present invention may be implemented;

Figure 2 depicts an example storage area network configuration in accordance with a preferred embodiment of the present invention;

Figure 3 is a block diagram illustrating a software configuration within a host computer in accordance with a preferred embodiment of the present invention;

Figure 4 is a flowchart illustrating communication between a device driver and a device loadbalance manager in accordance with a preferred embodiment of the present invention; and

Figure 5 is a flowchart illustrating the operation of a device loadbalance manager in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, **Figure 1** depicts a pictorial representation of a storage area network in which the present invention may be implemented. Storage area network (SAN) **100** contains SAN fabric **102**, which is a combination of interconnected switches, which collectively provide a routing infrastructure within SAN **100**.

In the depicted example, hosts **112**, **114** are connected to fabric **102** along with disk arrays **122**, **124**, **126**. Hosts **112**, **114** may be, for example, personal computers, network computers, servers, or the like. In the depicted example, hosts **112**, **114** access disk subsystems **122**, **124** and tape subsystem **126** through paths in the SAN fabric. SAN **100** may include additional hosts and/or other storage devices not shown. **Figure 1** is intended as an example, and not as an architectural limitation for the present invention.

Figure 2 depicts an example storage area network configuration in accordance with a preferred embodiment of the present invention. Host **210** is connected to a plurality of host bus adapters **212**, **214**, **216**, **218**. In the depicted example, the target devices are disk subsystem **222** and tape subsystem **224**. The disk subsystem and tape subsystem are connected to host bus adapters **242**, **244**, **246**, **248**. Host bus adapter **212** is connected to host bus adapter **242** through Fibre channel (FC) switch 1 **232**. Similarly, host bus adapter **214** is connected to host bus adapter **244** through FC switch 2 **234**, host bus

adapter **216** is connected to host bus adapter **246** through FC switch 3 **236**, and host bus adapter **218** is connected to host bus adapter **248** through FC switch 4 **238**.

The host and the storage subsystems are connected to the SAN fabric through four host bus adapters. Typically, a host or storage subsystem will be connected to between eight and thirty-two host bus adapters; however, more or fewer host bus adapters may be connected depending upon the implementation.

With interconnection between the switches and multiple levels of switches, the number of paths may become extensive. In addition, many of the paths share resources. Path control manager (PCM) software in host **210** for the disk subsystem selects a path for I/O to the disk subsystem. Similarly, a PCM for the tape subsystem selects a path for I/O to the tape subsystem.

In accordance with a preferred embodiment of the present invention, a mechanism is provided for monitoring I/O activity of each device and the total I/O activity for each adapter. When there is low I/O activity for the tape subsystem, the I/O for the disk subsystems may be spread across all available adapters and paths. When I/O activity for the tape subsystem increases, the I/O activity for the disk subsystems may be reduced on the adapter processing the tape I/O, but will continue across all other adapters. If the tape subsystem begins to report errors due to I/O starvation, the disk I/O activity may be adjusted until the errors stop.

Figure 3 is a block diagram illustrating a software configuration within a host computer in accordance with a

preferred embodiment of the present invention. Application layer **310** sends I/O operations for device driver layer **320**. In an exemplary embodiment, the device driver layer includes MPIO functionality. In the depicted example, the device driver layer includes device driver **330** for a first device, device driver **340** for a second device, and device driver **350** for a third device. More or fewer devices and, hence, more or fewer device drivers may be included. Device driver **330** includes path control manager (PCM) **332**; device driver **340** includes PCM **342**; and, device driver **350** includes PCM **352**. When I/O is to be sent to the storage subsystem, an appropriate one of PCMs **332**, **342**, **352** selects one of a plurality of paths as the transmission conduit.

Device loadbalance manager (DLM) **360** monitors for I/O activity for each device and the total activity per adapter. A device driver in device driver layer **320** for the tape subsystem, such as device driver **330**, monitors for errors. As the tape I/O activity increases and the tape subsystem begins to report errors due to I/O starvation, the device driver notifies the DLM of the errors and the DLM code begins to send commands to the PCMs, such as PCMs **342**, **352**, controlling the disk subsystem paths based on adapter I/O activity.

In response to the commands, PCMs **342**, **352** disable disk subsystem paths that are utilizing the adapter that the tape subsystem is using from being used for I/O. Disk subsystems with more alternate paths will be disabled first and disk subsystems with only one alternate path will be disabled last. The number of

paths to be disabled may also depend on I/O activity per disk.

Figure 3 is intended as an example and is not meant to limit the present invention. Modifications may be made to the software configuration within the scope of the present invention. For example, path management code may be embodied in an MPIO virtual device driver layer above device driver layer **320**. As another example, device loadbalancing path management code may be embodied in device driver layer **320**. Other modifications will be apparent to those of ordinary skill in the art.

Figure 4 is a flowchart illustrating communication between a device driver and a device loadbalance manager in accordance with a preferred embodiment of the present invention. The process begins and the device driver registers callback routines with the DLM (step **402**). Then, the DLM registers routines with the device driver (step **404**) and the process ends. The device driver may then use the routines to report I/O returns to the DLM. Similarly, the DLM may use the callback routines to send commands to the PCMs in the device drivers.

Turning now to **Figure 5**, a flowchart is shown illustrating the operation of a device loadbalance manager in accordance with a preferred embodiment of the present invention. The process begins and the device driver sends an I/O (step **502**) and the I/O returns from the device (step **504**). The device driver calls into the DLM with the I/O results (step **506**).

Next, a determination is made as to whether the tape subsystem is under run (step **508**). A tape subsystem is

under run if the tape subsystem does not receive enough I/O to keep the tape spinning. This determination may be made by determining whether I/O activity for the tape subsystem drops below a predetermined threshold. Alternatively, this determination may be made by determining whether an I/O starvation error is received from the tape subsystem. In yet another embodiment, a combination of these determinations may be used.

If the tape subsystem is under run, the DLM calls into the device driver to reduce the devices using the same path as the tape (step **510**). The DLM may send commands instructing the PCMs for one or more disk subsystems to reduce the priority of paths using the host bus adapter of the tape subsystem or to disable the paths altogether. Thereafter, the DLM updates activity statistics (step **512**) and ends.

If the tape subsystem is not under run in step 508, a determination is made as to whether the tape subsystem is over run (step **514**). A tape subsystem is over run if the tape subsystem receives too much I/O. This determination may be made by determining whether I/O activity for the tape subsystem exceeds a predetermined threshold. Alternatively, this determination may be made by determining whether a buffer overflow error is received from the tape subsystem. In yet another embodiment, a combination of these determinations may be used.

If the tape subsystem is over run, a determination is made as to whether disk activity is high for one or more of the disk subsystems (step **516**). If disk activity

is high, the DLM calls into the device driver to increase the devices using the same path as the tape (step **518**). The DLM may send commands instructing the PCMs for one or more disk subsystems to increase the priority of paths using the host bus adapter of the tape subsystem or to enable the paths that were previously disabled. Thereafter, the DLM updates activity statistics (step **512**) and ends.

If the tape subsystem is not over run in step **514** or the disk activity is not high in step **516**, the process continues to step **512** to update activity statistics and the process ends.

Preferably, the DLM balances the load across the adapters while keeping the HBA for the tape subsystem fairly dedicated to the tape subsystem. For example, if activity for a disk subsystem is high while activity for the tape subsystem is very low, the DLM may send commands to allow the disk subsystem to use paths that share the HBA of the tape subsystem. On the other hand, if I/O activity for a first disk subsystem is very high and I/O activity for a second disk subsystem is not very high, the DLM may send commands to allow the second disk subsystem to use paths that share the HBA of the tape subsystem; however, the DLM may not send commands to allow the first disk subsystem to use paths that use the HBA of the tape subsystem, because the high activity for the first disk subsystem may result in starvation errors in the tape subsystem.

Thus, the present invention solves the disadvantages of the present invention by providing a device

loadbalance manager for monitoring I/O activity of each device and the total I/O activity for each adapter. When there is low I/O activity for the tape subsystem, the device loadbalance manager spreads the I/O for the disk subsystems across all available adapters and paths. When I/O activity for the tape subsystem increases, the device loadbalance manager may reduce the I/O activity for the disk subsystems on the adapter processing the tape I/O, while allowing I/O activity to continue across all other adapters. If the tape subsystem begins to report errors due to I/O starvation, the disk I/O activity may be adjusted until the errors stop.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions. The computer readable media may take the form of coded

formats that are decoded for actual use in a particular data processing system.

The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.